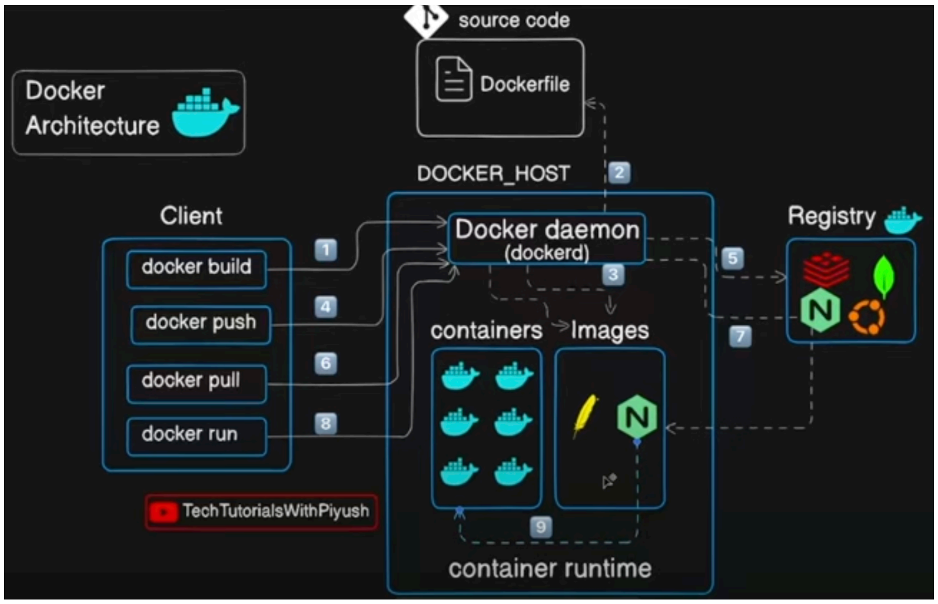
Kubernetes Class

Day 0/40 - Certified Kubernetes Administrator Course Introduction - CKA Full Course 2025 ✅

- No notes

Day 1/40 - Docker Tutorial For Beginners - Docker Fundamentals - CKA Full Course 2025 ✅

- Docker: Lightweight sandbox environment



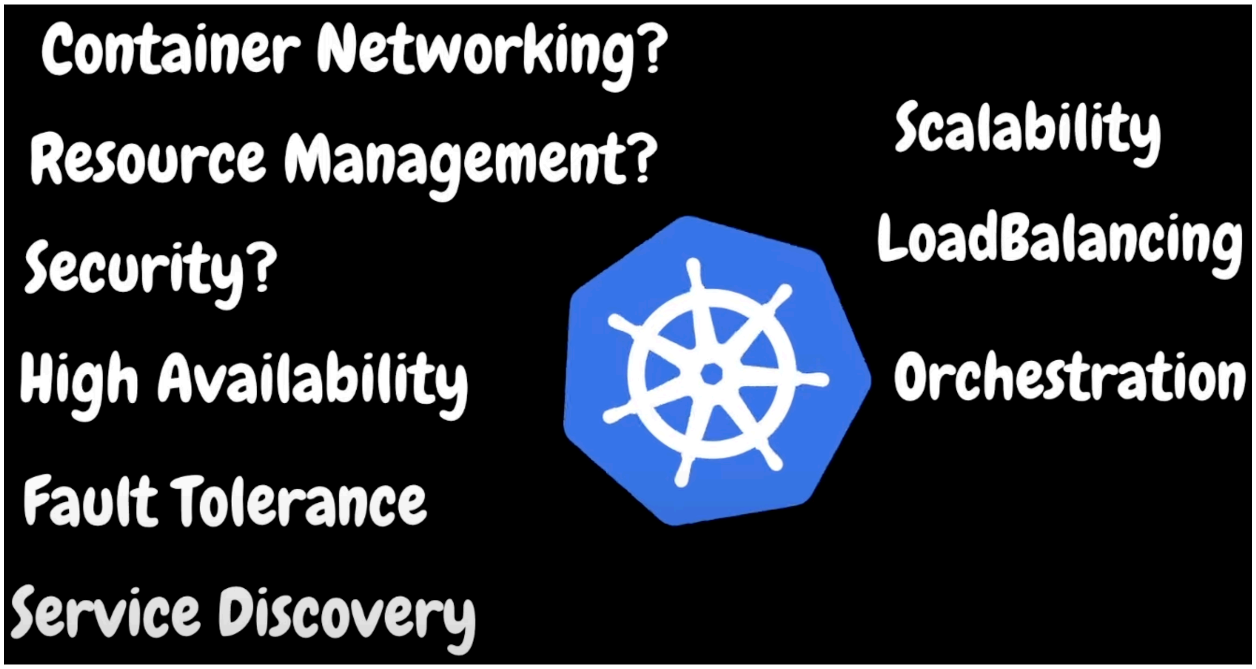
Day 2/40 - How To Dockerize a Project - CKA Full Course 2025 ✅

```
# Nice Docker commands
docker build -t day02-todo .

docker tag day02-todo:latest orbit196/test-repo:latest
docker push orbit196/test-repo:latest
docker pull orbit196/test-repo:latest

docker run -dp 3000:3000 orbit196/test-repo:latest

docker exec -it <container_name> sh
```

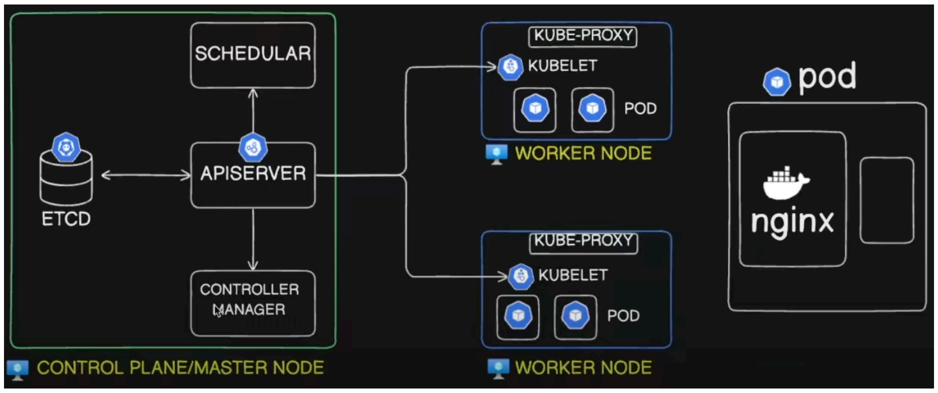
Day 3/40 - Multi Stage Docker Build - Docker Tutorial For Beginners - CKA Full Course 2024 ✅

```
# Nice Docker commands
docker build -t multi-stage .
docker image rm orbit196/test-repo

docker ps
docker run -it -dp 3000:3000 multi-stage

docker logs <CONTAINER ID>
docker exec -it <CONTAINER ID> sh
docker inspect <CONTAINER ID>
```

Day 4/40 - Why Kubernetes Is Used - Kubernetes Simply Explained - CKA Full Course 2025 ✅



Day 5/40 - What is Kubernetes - Kubernetes Architecture Explained ✅

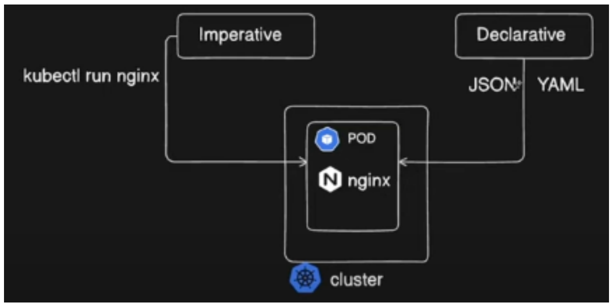


Day 6/40 - Kubernetes Multi Node Cluster Setup Step By Step | Kind Tutorial ✅

- **Kubectl Quick Reference**
- **Kubernetes Blog**

```
# Create kind cluster
kind create cluster --image kindest/node:v1.32.2@sha256:f226345927d7e348497136874b6d207e0b32cc52154ad8
kind create cluster --image kindest/node:v1.32.2@sha256:f226345927d7e348497136874b6d207e0b32cc52154ad8

# Check the cluster
kubectl cluster-info --context kind-cka-cluster1

# Kubectl commands
kubectl version --client
kubectl get nodes
kubectl get nodes -o wide
kubectl config get-contexts
kubectl config use-context kind-cka-cluster1

# Kind commands
kind get clusters
```

[Day 7/40 - Pod In Kubernetes Explained | Imperative VS Declarative Way | YAML Tutorial]() ✅ **(HW)***

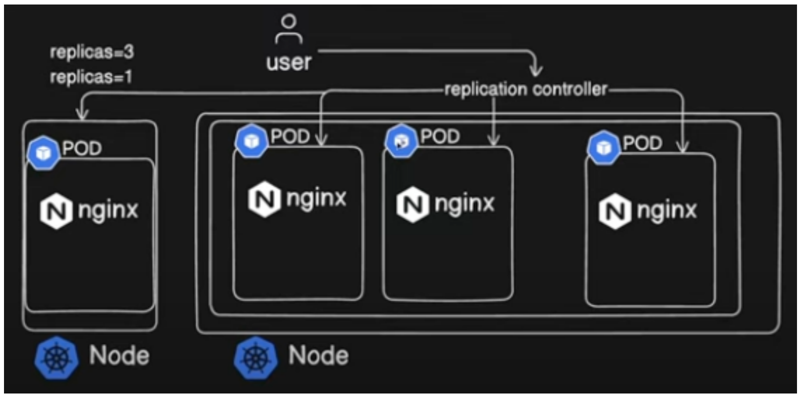


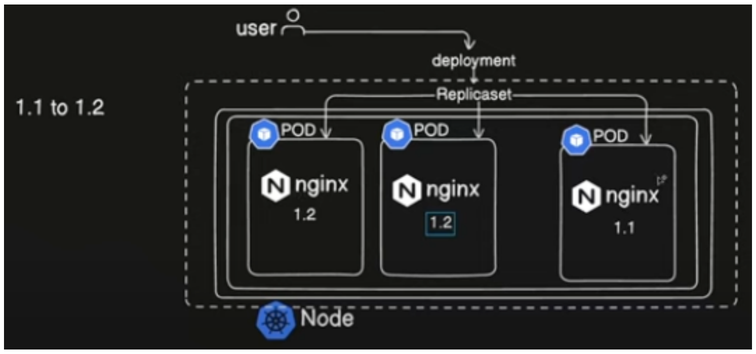
The **pod** four top-level fields are:

- apiVersion
- kind
- metadata
- spec

```
# Kubectl pod commands
kubectl run nginx-pod --image=nginx:latest # The imperative way!
kubectl get pods
kubectl explain pod
kubectl create -f pod.yaml
kubectl delete pod nginx-pod
kubectl apply -f pod.yaml
kubectl describe pod nginx-pod # Detailed description of specific pods!
kubectl edit pod nginx-pod # Don't have to apply on the pod again!
kubectl exec -it nginx-pod -- sh # Get inside the pod!
kubectl get pods nginx-pod --show-labels
kubectl get pods -o wide

# Dry run
kubectl run nginx --image=nginx --dry-run=client -o yaml > pod-new.yaml
kubectl run nginx --image=nginx --dry-run=client -o json > pod-new.json
```

Day 8/40 - Kubernetes Deployment, Replication Controller and ReplicaSet Explained ✅ **(HW)***





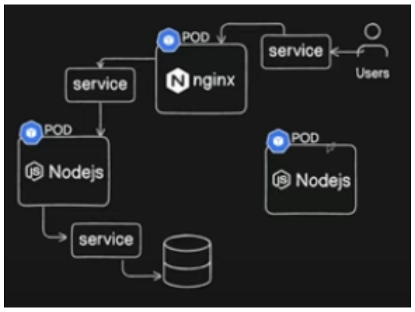
- **Deployment** created the **ReplicaSet** which created the **Pods**!

```
# Kubectl Replication Controller and ReplicaSet commands
kubectl explain rc
kubectl get rc
kubectl describe pod nginx-rc-hckfz
kubectl delete rc/nginx-rc # Deleting the Replication Controller
kubectl delete rs/nginx-rs # Deleting the ReplicaSet
kubectl edit rs/nginx-rs
kubectl scale --replicas=10 rs/nginx-rs
kubectl scale --help

# Kubectl Deployment commands
```

```
kubectl get deploy
kubectl set image deploy/nginx-deploy \
nginx=nginx:1.9.1
kubectl describe deploy/nginx-deploy
kubectl rollout history deploy/nginx-deploy
kubectl rollout undo deploy/nginx-deploy
kubectl create deploy deploy/nginx-new --image=nginx --dry-run=client -o yaml > deploy.yaml

# Get all the objects running in the cluster
kubectl get all
```

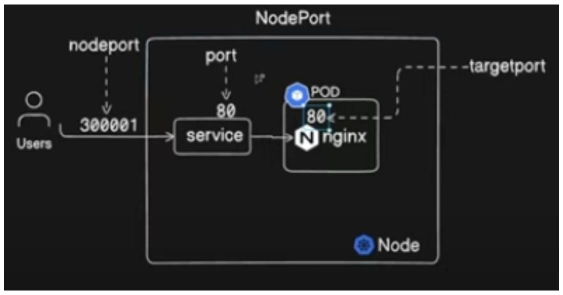
[Day 9/40 - Kubernetes Services Explained - ClusterIP vs NodePort vs Loadbalancer vs External]() ✅ **(HW)***

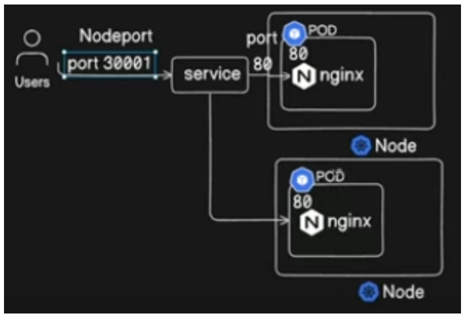
```
# Kubectl Service commands
kubectl explain service
kubectl get pod --show-labels
kubectl create -f nodeport.yaml
kubectl get svc
kubectl get pod -o wide
kind create cluster --config=kind.yaml --name=cka-cluster3 # Same command as below!
kind create cluster --config kind.yaml --name cka-cluster3 # Same command as above!
kind delete cluster --name cka-cluster3
kubectl get nodes
kubectl apply -f rc.yaml
kubectl apply -f nodeport.yaml
kubectl get svc
kubectl describe svc nodeport-svc
kubectl describe pod nginx-deploy-7fff95c694-gbdps # IP: 10.244.1.2
kubectl delete pod nginx-deploy-7fff95c694-gbdps
kubectl describe pod nginx-deploy-7fff95c694-cg8vn # IP: 10.244.1.3
kubectl apply -f clusterip.yaml
kubectl get svc
kubectl describe svc/cluster-svc # Same command as below!
kubectl describe svc cluster-svc # Same command as above!
kubectl get ep
kubectl create -f lb.yaml
kubectl get svc
```

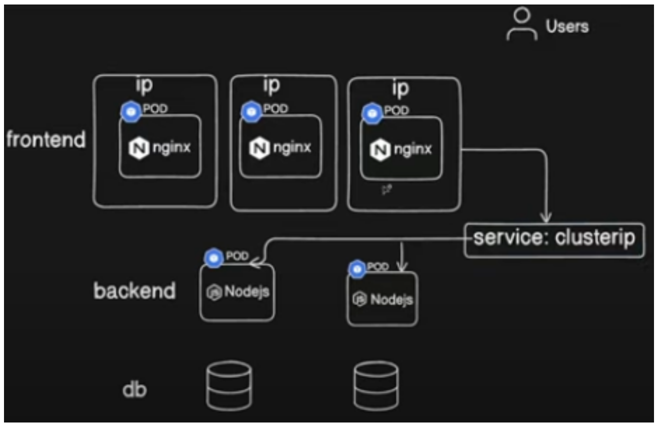


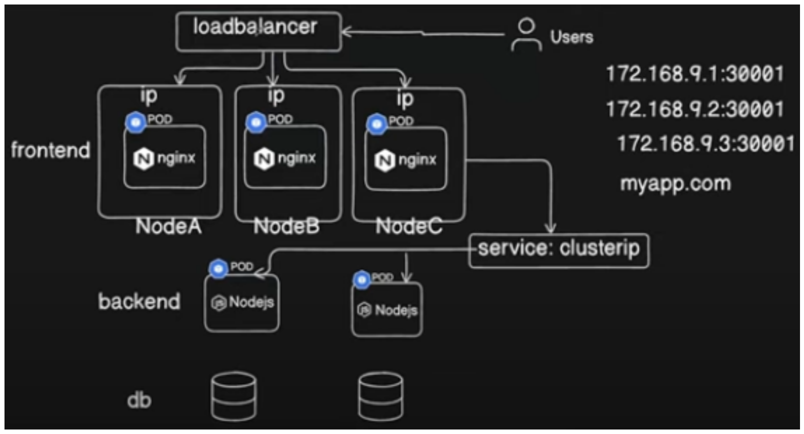
Kubernetes **services**:

- ClusterIP
- NodePort
- ExternalName
- LoadBalancer

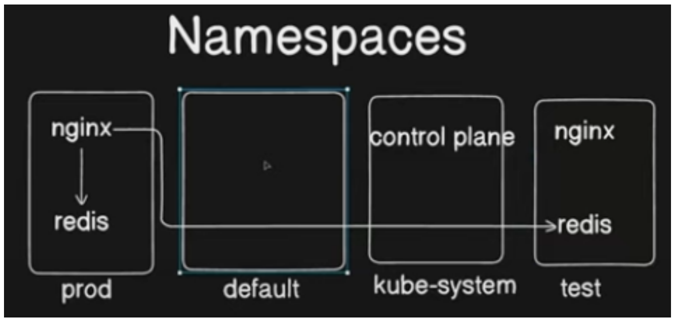




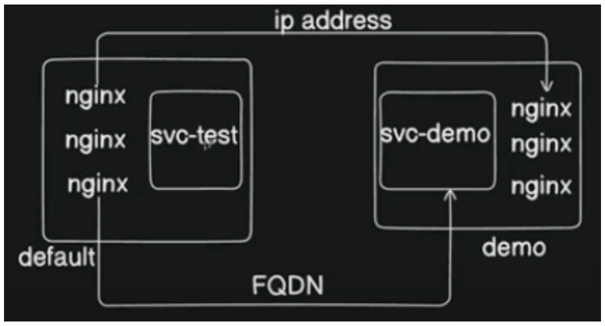




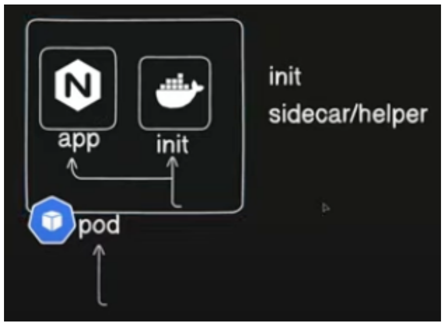
Day 10/40 - Kubernetes Namespace Explained - CKA Full Course 2025 ✅



```
kubectl get all --namespace=kube-system # Control plane components
kubectl get all -n kube-system
kubectl get all -n=default # Get information from the default namespace
kubectl apply -f ns.yaml
kubectl get ns
kubectl delete ns/demo
kubectl create ns demo # Create namespace imperatively
kubectl create deploy nginx-demo --image=nginx -n demo
kubectl get deploy
kubectl create deploy nginx-test --image=nginx
kubectl get deploy
kubectl exec -it nginx-demo-87cd4cbb7-hsk29 -n demo -- sh # Namespace: demo
kubectl exec -it nginx-test-b548755db-vnlvx -- sh        # Namespace: default
kubectl get pods -n=demo -o wide # Namespace: demo; IP Address: 10.244.1.3
kubectl get pods -o wide        # Namespace: default; IP Address: 10.244.1.4
kubectl scale --replicas=3 deploy/nginx-demo -n=demo
kubectl scale --replicas=3 deploy/nginx-test
kubectl expose deploy/nginx-demo --name=svc-demo --port 80 -n=demo # IP Address: 10.96.33.205
kubectl get svc -n=demo
kubectl expose deploy/nginx-test --name=svc-test --port 80        # IP Address: 10.96.132.194
kubectl get svc
kubectl get pods -n demo
kubectl exec -it nginx-demo-87cd4cbb7-hsk29 -n demo -- sh
kubectl get pods
kubectl exec -it nginx-test-b548755db-cp62h -- sh
# We are ale to reach the pods and services from the different namespaces with
# their IP Addresses, but not from their Hostname
curl svc-test
curl svc-demo
cat /etc/resolv.conf # demo.svc.cluster.local
                    # default.svc.cluster.local
# To access outside the namespace, use FQDN
curl svc-test.default.svc.cluster.local
curl svc-demo.demo.svc.cluster.local

# Key point: Hostnames are not cluster-wide, they are namespace-wide!
```



Day 11/40 - Multi Container Pod Kubernetes - Sidecar vs Init Container ✅ **(HW; Video Again)***



```
kubectl create -f pod.yaml
k delete pod myapp
k create -f pod.yaml
k logs pod/myapp
k logs pod/myapp -c init-myservice
k create deploy nginx-deploy --image nginx --port 80
k get deploy
k expose deploy nginx-deploy --name myservice --port 80
k logs pod/myapp -c init-myservice
```

```
k get pod
k exec -it myapp -- printenv
k exec -it myapp -- sh
k delete pod myapp
k apply -f pod.yaml
k create deploy mydb --image redis --port 80
k expose deploy mydb --name mydb --port 80
k get pod -w
```

Day 12/40 - Kubernetes Daemonset Explained - Daemonsets, Job and Cronjob in Kubernetes